



Estruturas de Controle em Python

Bem-vindo ao nosso estudo sobre a Estruturas de Controle em Python. Este estudo ajudará você a desenvolver programas na linguagem Python com as principais estruturas de controle, dando uma boa base para você criar programas de Inteligência Artificial, mais à frente. Bons estudos!

Estrutura Sequencial

Uma estrutura sequencial é aquela em que o código é executado a cada linha de comando, sem sofrer nenhum desvio. Portanto, o código é executado desde a primeira linha até o final, passando pela entrada, processamento e saída. Segue abaixo um exemplo de estrutura sequencial.

```
1 # Estrutura sequencial:
2 # Entrada:
3 num1 = int(input('Digite um número: '))
4 num2 = int(input('Digite outro número: '))
5 # Processamento:
6 soma = num1 + num2
7 sub = num1 - num2
8 # Saída:
9 print(num1,'+',num2,'=',soma)
10 print(num1,'-',num2,'=',sub)
```

O exemplo acima faz a entrada de dois números inteiros (linhas 3 e 4), processa a soma e a subtração desses números (linhas 6 e 7) e mostra os resultados (linhas 9 e 10). Veja que para a entrada dos dados foi utilizado o comando input. Como os números seriam armazenados nas variáveis num1 e num2 como inteiros, antes de serem guardados r respectivas variáveis, foram convertidos para inteiro através da função int.

Resultado do processamento:



```
Digite um número: 10
Digite outro número: 3
10 + 3 = 13
10 - 3 = 7
```

Essa é uma estrutura sequencial, pois é executada sequencialmente, do início ao fim.

Estruturas Condicionais

Em algumas situações, o fluxo de execução do programa necessita ser desviado ou alguma condição necessita ser testada. Dessa forma, as Estruturas Condicionais permitem a escolha de um grupo de ações a ser executado quando determinadas condições, representadas por expressões lógicas ou relacionais, sejam ou não satisfeitas. Uma estrutura condicional é aquela que testa uma condição para verificar se deve executar um bloco de comandos ou outro. A estrutura condicional pode ser simples, composta ou aninhada.

Estrutura condicional simples

A estrutura condicional simples é aquela que só executa determinados comandos quando a condição for verdadeira, se for falsa, não faz nada. Veja o exemplo abaixo.

```
primeiro = float(input('Digite um número inteiro: '))
segundo = float(input('Digite outro número: '))
# Estrutura condicional simples:
if primeiro > segundo:
    aux = primeiro
    primeiro = segundo
    segundo = aux
print('Números em ordem crescente: ', primeiro, segundo)
```

No exemplo, o programa recebe dois números e mostra-os em ordem crescente. Então, ele verifica se o primeiro número é maior que o segundo. Se for, ele inverte os valores. Se não for, não precisa fazer nada porque já está em ordem.

A indentação (tabulação) é muito importante no código Python. Veja que os comandos que estão dentro da estrutura do if estão indentados. Isso é obrigatório na linguagem Python.

Estrutura Condicional Composta

Na estrutura condicional composta se a condição verificada for verdadeira, alguns comandos são acionados, caso contrário são acionados outros comandos. Veja no exemplo abaixo.

```
numero = int(input('Digite um número inteiro: '))
if numero % 2 == 0:
    print(numero, 'é par')
else:
    print(numero, 'é impar')
```

No exemplo, é digitado um número inteiro e na estrutura condicional é verificado se o número é divisível por 2. Se for, mostra que o número é par, caso contrário, mostra que o número é ímpar.

Estrutura Condicional Aninhada

Na estrutura condicional aninhada, existem mais de duas condições a serem testadas. Veja o exemplo abaixo.

```
numero = float(input('Digite um número: '))
# estrutura condicional aninhada:
if numero > 0:
    print(numero, 'é positivo')
elif numero < 0:
    print(numero, 'é negativo')
else:
    print(numero, 'é zero')
```

Neste exemplo, é verificado se o número digitado é positivo, negativo ou zero. Vimos, por esse exemplo, que as condições vão sendo testadas através dos comandos if e elif (senão / se). E a última condição fica no else.

Estruturas de Repetição

As estruturas de repetição são utilizadas quando precisamos repetir um bloco de comandos dentro de nosso programa. Na linguagem Python existem duas estruturas de repetição: for e while.

Estrutura for:

A estrutura for é utilizada, geralmente, quando sabemos de antemão quantas vezes serão executados os comandos. Existem três formas de utilização do for. Veja os exemplos abaixo.

```
# Primeiro Exemplo de for

x = 0

for x in range (3) :    # x vai de 0 a 2
       print ('O valor de x é: %d' % x);
print ('Saiu do laço')
```

```
# Segundo exemplo de for

for x in range (50, 100) : # x vai de 50 a 99
    if x == 88 :
        break              # se x for igual a 88, sai do laço
    print(x)
print('Saiu do for')
```



```
# Terceiro exemplo com for - soma dos números impares
total = 0
numero = int(input('Digite um número: '))
if (numero % 2) == 0:
    numero = numero -1
for i in range(numero,0,-2):    # i vai de numero até 0, decrementando de 2
    total = total + i
    print('Valor de i %d:' % i)
print("A soma dos números impares é %d " % total)
```

No primeiro exemplo, a estrutura **for x in range(3):**, o que está indentado será executado três vezes, pois o valor de x vai de 0 a 2, incrementando de 1 em 1.

No segundo exemplo, a estrutura **for x in range (50, 100):**, o x vai de 50 a 99, incrementando de 1 em 1. Dentro do laço tem uma condição verificando se x = 88. Se for, o laço é interrompido pelo comando break.

No terceiro exemplo, a estrutura **for i in range(numero, 0, -2):**, a variável de controle i inicia no valor do número e vai até 1, decrementando de 2 em 2.

Exemplo de for com lista:

```
lista = []
soma = 0
for i in range(5):
    lista.append(int(input('Digite um número: ')))
# Somando os números:
for e in lista:
    soma = soma + e
# mostrando o resultado:
print('Lista:', lista)
print('Soma dos números:', soma)
```

Nesse exemplo, no primeiro for são digitados 5 números e armazenados na lista. No segundo for os números são somados e o resultado guardado em soma. V[illegible] que o segundo for é diferente, é específico para ser usado com coleções, como a lista. É chamado de for aprimorado.

Estrutura while:

A estrutura de repetição while pode ser utilizada quando o programa precisa testar determinada condição antes de executar um conjunto de comandos repetidas vezes. Se a condição avaliada for verdadeira, o conjunto de comandos dentro da estrutura de repetição é executado e após esta execução, a condição é novamente avaliada. Se o resultado da avaliação for falso, este conjunto de comandos não será executado e o fluxo do programa segue normalmente. Nesta estrutura de repetição, pode ocorrer do conjunto de comando não ser executado nenhuma vez. Veja abaixo um exemplo de estrutura while.

```
#Exemplo while

x = 0
while x < 3 :
    print('O valor de x é: %d' % x)
    x = x + 1
print('Saiu do while')
```

Pelo exemplo, vimos que se houver uma variável de controle (neste caso é o x), é necessário incrementá-la dentro do laço. Nessa estrutura de repetição, nem sempre sabemos quantas vezes os comandos serão executados.

Atividade Extra

Para se aprofundar no assunto desta aula, pesquise artigos e vídeos sobre estruturas de controle na linguagem Python. Como sugestão, leia os capítulos referentes ao assunto no livro Introdução à Programação com Python, de Nilo Ney Coutinho Menezes.

Referência Bibliográfica

MENEZES, Nilo N. C. Introdução à Programação com Python: Algoritmos e lógica de programação para iniciantes. São Paulo: Novatec, 2014.

SUMMERFIELD, Mark. Programação em Python 3: uma introdução completa à linguagem Python. Rio de Janeiro: Alta Books, 2012.

Ir para exercício